Opérateur DEFI_LIST_INST

1 But

Définir la liste d'instants de calcul, ainsi que sa gestion pour les algorithmes de résolution itératifs (commandes STAT_NON_LINE et DYNA_NON_LINE) :

- possibilités de re-découpage du pas de temps en cas d'échec,
- possibilités d'adaptation du pas de temps.

Produit une structure de données list_inst.

Table des Matières

2 Syntaxe

```
deflist[list_inst]  =  DEFI_LIST_INST(

# 1) mots-clés pour la définition a priori de la liste d'instant

DEFI_LIST =   _F(

   ♦  METHODE       =  /'MANUEL',                         [DEFAUT]
                       /'AUTO',

   ♦  /  LIST_INST    =  list,                               [l_r8]
      /  VALE         =  vale,                               [R]
      /  ♦  RESULTAT  =  resu,                               [resultat]
         ♦  SUBD_PAS  =  npas                                [I]

   # si METHODE = 'AUTO'
       ◊ PAS_MINI     =   pasmin,                         [R]
       ◊ PAS_MAXI     =   pasmax,                         [R]
      ◊  NB_PAS_MAXI  =  /  1000000                       [DEFAUT]
                         /  nbpasmax,                     [I]
                )
# 2) mots-clés pour le comportement en cas d'échec

    ♦  ECHEC =   _F(
      ◊  EVENEMENT =  /'ERREUR',                          [DEFAUT]
                      /'DIVE_RESI',
                      /'DELTA_GRANDEUR'
                      /'COLLISION'
                      /'INTERPENETRATION'
                      /'INSTABILITE'


      # Si EVENEMENT = 'DELTA_GRANDEUR'
      ♦  VALE_REF  =  valref,                             [R]
       ♦ NOM_CHAM  =   /'DEPL' ,
                      /'SIEF_ELGA',
                      /'VARI_ELGA',
      ♦  NOM_CMP   =  cmp ,                               [TXT]
      ◊  ACTION    =  /'DECOUPE',                         [DEFAUT]
                      /'ARRET',
         # Si ACTION = 'DECOUPE'
         ◊  SUBD_METHODE    =  /'MANUEL',                 [DEFAUT]
                               /'AUTO',
            # Si SUBD_METHODE = 'MANUEL'
            ◊  SUBD_PAS        =  4.,                     [DEFAUT]
                                  pas,                    [R]
            ◊  SUBD_NIVEAU     =  3 ,                     [DEFAUT]
                                  niv,                    [I]
            ◊  SUBD_PAS_MINI   =  0.,                     [DEFAUT]
                                  pasmin,                 [R]
            # Si SUBD_METHODE   = ' AUTO '
            ◊  SUBD_PAS_MINI   =  0 ,                     [DEFAUT]
                                  pasmin,                 [R]


      # Si EVENEMENT = 'ERREUR'
      ◊  ACTION    =  /'DECOUPE',                         [DEFAUT]
                      /'ARRET',
                      /'ITER_SUPPL',
```

```
                    /'AUTRE_PILOTAGE',

        # Si ACTION = 'DECOUPE'/'AUTRE_PILOTAGE'
        ◊  SUBD_METHODE    =  /'MANUEL',           [DEFAUT]
                              /'AUTO',
           # Si SUBD_METHODE = 'MANUEL'
           ◊  SUBD_PAS        =  4 .,                  [DEFAUT]
                                 pas,              [R]
           ◊  SUBD_NIVEAU     =  3 ,               [DEFAUT]
                                 niv,              [I]
           ◊  SUBD_PAS_MINI   =  0 .,                  [DEFAUT]
                                 pasmin,           [R]
           # Si SUBD_METHODE = ' AUTO '
           ◊  SUBD_PAS_MINI   =  0 ,               [DEFAUT]
                                 pasmin,           [R]


        # Si ACTION = 'ITER_SUPPL '
        ◊  PCENT_ITER_PLUS  =  50 ,                    [DEFAUT]
                              pcent,               [I]
        ◊  SUBD_METHODE    =  /'MANUEL',           [DEFAUT]
                              /'AUTO',
           # Si SUBD_METHODE = 'MANUEL'
           ◊  SUBD_PAS        =  4.,               [DEFAUT]
                                 pas,              [R]
           ◊  SUBD_NIVEAU     =  3 ,               [DEFAUT]
                                  niv,             [I]
           ◊  SUBD_PAS_MINI   =  0.,               [DEFAUT]
                                 pasmin,           [R]
           # Si SUBD_METHODE = ' AUTO '
           ◊  SUBD_PAS_MINI   =  0 ,               [DEFAUT]
                                 pasmin,           [R]


     # Si EVENEMENT = ' COLLISION '
     ◊  ACTION    =  /'DECOUPE',                   [DEFAUT]
                     /'ARRET',
        # Si ACTION = 'DECOUPE'
        ◊  SUBD_METHODE    =  /'AUTO',             [DEFAUT]
                              /'MANUEL',
           # Si SUBD_METHODE = 'MANUEL'
           ◊  SUBD_PAS        =  4.,               [DEFAUT]
                                 pas,              [R]
           ◊  SUBD_NIVEAU     =  3 ,               [DEFAUT]
                                  niv,             [I]
           ◊  SUBD_PAS_MINI   =  0 .,                 [DEFAUT]
                                 pasmin,           [R]
           # Si SUBD_METHODE = ' AUTO '
              ♦  SUBD_ INST   =  subd_inst,        [R]
              ♦  SUBD_ DUREE  =  SUBD_DUREE,       [R]


     # Si EVENEMENT = ' INTERPENETRATION '
      ♦ PENE_MAXI =    penmax ,                    [R]
     ◊  ACTION    =  /' ADAPT_COEF_PENA ',         [DEFAUT]
                     /'ARRET',
        # Si ACTION = ' ADAPT_COEF_PENA '
        ◊  COEF_MAXI    =  1E12                    [DEFAUT]
                           coemax ,                [R]
```

```
        # Si EVENEMENT = ' DIVE_RESI '                   [R]
        ◊  ACTION       =  /' DECOUPE ',                 [DEFAUT]
        ◊  SUBD_METHODE =  /'MANUEL',                        [DEFAUT]
                           /'AUTO',
               # Si SUBD_METHODE = 'MANUEL'
               ◊  SUBD_PAS        =  4.,                     [DEFAUT]
                                     pas,                    [R]
               ◊  SUBD_NIVEAU     =  3 ,                     [DEFAUT]
                                     niv,                    [I]
               ◊  SUBD_PAS_MINI   =  0.,                     [DEFAUT]
                                     pasmin,                 [R]
               # Si SUBD_METHODE = ' AUTO '
               ◊  SUBD_PAS_MINI   =  0 ,                     [DEFAUT]
                                     pasmin,                 [R]


      # Si EVENEMENT = 'INSTABILITE'
      ◊  ACTION    =  /'CONTINUE',                           [DEFAUT]
                      /'ARRET',
```

3) mots-clés pour l'adaptation (si `METHODE = 'AUTO'`)

```
  ◊  ADAPTATION   =  _F(

      ◊  EVENEMENT    =  /'SEUIL',                           [DEFAUT]
                         /'TOUT_INST'
                         /'AUCUN'

      # si EVENEMENT = 'SEUIL'
       ◊ NB_INCR_SEUIL       =  /2 ,                         [DEFAUT]
                                /nbincseuil ,                [I]
       ◊ NOM_PARA            =  'NB_ITER_NEWTON'             [DEFAUT]
       ◊ CRIT_COMP           =  /'LE',                       [DEFAUT]
                             =  /'LT',
                             =  /'GE',
                             =  /'GT',
      ◊  VALE_I              =   nbvalseuil,                 [I]
      ◊  MODE_CALCUL_TPLUS   =  /'FIXE',                     [DEFAUT]
                                /'DELTA_GRANDEUR'
                                /'ITER_NEWTON'
                                /'IMPLEX'

         # si MODE_CALCUL_TPLUS  = 'FIXE'
          ◊ PCENT_AUGM          =  /100.,                    [DEFAUT]
                                   / pcent ,                 [R]

         # si MODE_CALCUL_TPLUS  = 'DELTA_GRANDEUR'
          ♦ VALE_REF            =  valref ,                  [R]
          ♦ NOM_CHAM            =  /'DEPL' ,
                                   /'SIEF_ELGA',
                                   /'VARI_ELGA',
          ♦ NOM_CMP             =  cmp ,                     [TXT]

         # si MODE_CALCUL_TPLUS = 'ITER_NEWTON'
          ♦ NB_ITER_NEWTON_REF  =    nitref ,                [I]

                        )

  ◊  INFO      =  /  1,                                      [DEFAUT]
                  /  2,                                      [I]
```

```
)
```

3 Opérandes

3.1 Mot-clé DEFI_LIST

3.1.1 Opérande METHODE

```
♦  METHODE      =  /'MANUEL',              [DEFAUT]
                   /'AUTO',
```

Cet opérande permet de choisir le mode de gestion de la liste d'instants.

Par défaut, la gestion est manuelle (METHODE='MANUEL'), ce qui signifie que l'on va parcourir exactement la liste d'instants définie par l'utilisateur (voir §3.1.2). En cas de sous-découpage, de nouveaux instants peuvent être insérés.

En gestion automatique (METHODE='AUTO'), le code calcule automatiquement les nouveaux instants de calcul. Les modes de calcul des nouveaux pas de temps sont définis sous le mot-clé ADAPTATION. Les instants de calculs définis par l'utilisateur (voir §3.1.2) sont cependant respectés.

3.1.2 Opérandes LIST_INST et VALE

```
♦  /  LIST_INST          =  list,              [l_r8]
   /  VALE               =  vale,              [R]
   /  ♦  RESULTAT        =  resu,              [resultat]
      ♦  SUBD_PAS        =  npas               [I]
```

Les instants de calcul peuvent être définis de 3 manières différentes :

- soit en renseignant le mot-clé LIST_INST avec un concept défini au préalable par l'opérateur DEFI_LIST_REEL [U4.34.01],
- soit en renseignant le mot-clé VALE avec une liste de réels,
- soit en renseignant le mot-clé RESULTAT avec un concept produit par STAT_NON_LINE ou DYNA_NON_LINE. Ce mot-clé permet de raffiner une liste d'instants déjà existante. En effet, on définit alors une liste d'instants npas **fois plus fine** que la liste des instants réellement calculés dans le STAT_NON_LINE ou DYNA_NON_LINE. Il est à noter que cette liste d'instants réellement calculés peut être différente de la liste des instants donnée en entrée du STAT_NON_LINE ou DYNA_NON_LINE (par exemple à cause du sous-découpage) et de la liste des instants archivés (si on n'archive pas tous les instants). Cette fonctionnalité est utile pour vérifier l'indépendance des résultats à la discrétisation temporelle. Par exemple, pour obtenir une liste d'instants 2 fois plus fine, on choisit SUBD_PAS=2.

Attention ! Si la SD résultat resu provient d'un calcul avec reprise, il n'est pas possible de la réutiliser dans la commande car on ne peut extraire de liste d'instants uniques (certains instants seront en double).

3.1.3 Opérandes PAS_MINI, PAS_MAXI, NB_PAS_MAXI

```
◊  PAS_MINI     =  pasmin,               [R]
◊  PAS_MAXI     =  pasmax,               [R]
◊  NB_PAS_MAXI  =  nbpasmax,             [I]
```

Ces opérandes ne sont à renseigner qu'en cas de gestion automatique de la liste des instants (METHODE ='AUTO').

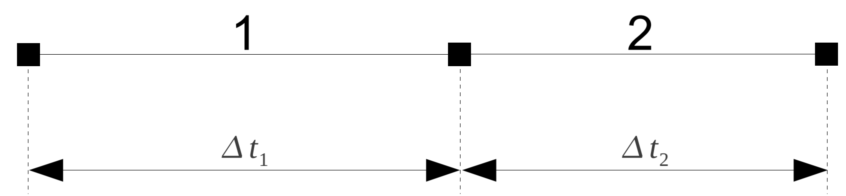


Figure 3-1: Adaptation automatique du pas de temps

L'opérande `PAS_MINI` permet de spécifier le pas de temps minimal. Si au cours de l'adaptation du pas de temps, on est amené à traiter un pas de temps inférieur à `pasmin`, alors le calcul s'arrête. Si cet opérande n'est pas renseigné, dans le cas général `PAS_MINI` vaut alors 10^{-12} (c'est d'ailleurs la valeur minimale autorisée) ; s'il n'est pas renseigné et que `MODE_CALCUL_TPLUS='IMPLEX'`, alors le `PAS_MINI` vaut 1/1000ème du premier pas de calcul (forcément entré par l'utilisateur *via* `LIST_INST`).

Si $\Delta t_2 < \text{PAS_MINI} \;:\; \rightarrow STOP$

Attention, la vérification de `PAS_MINI` n'est **pas** effectué pendant le sous-découpage éventuel du pas de temps en cas d'échec (c'est `SUBD_PAS_MINI` qui agit pendant le sous-découpage). L'opérande `PAS_MINI` n'agit **que** pour le calcul d'un nouveau pas de temps.

L'opérande `PAS_MAXI` permet de spécifier le pas de temps maximal, mais son fonctionnement est différent de celui de `PAS_MINI`. En effet, si au cours de l'adaptation du pas de temps, on est amené à traiter un pas de temps supérieur à `pasmax`, alors le calcul ne s'arrête pas (contrairement au fonctionnement de `PAS_MINI`), mais le pas de temps est alors limité à `pasmax`. Si cet opérande n'est pas renseigné, on ne fera pas cette étape ; s'il n'est pas renseigné et que `MODE_CALCUL_TPLUS='IMPLEX'`, alors le `PAS_MAXI` vaut 10 fois le premier pas de calcul (forcément entré par l'utilisateur *via* `LIST_INST`).

Si $\Delta t_2 > \text{PAS_MAXI} \;:\; \rightarrow \Delta t_2 = \text{PAS_MAXI}$

L'opérande `NB_PAS_MAXI` permet de spécifier le nombre de pas de temps au-delà duquel le calcul s'arrête. La valeur par défaut est un million (c'est aussi la valeur maximale autorisée).

3.2 Mot-clé `ECHEC`

Ce mot-clef facteur répétable permet de gérer la liste d'instants en cas d'échec de convergence. À chaque occurrence du mot-clé `ECHEC` , on définit une cause d'échec (voir § 3.2.1) et une action à effectuer si cette cause est satisfaite (voir § 3.2.2). Les actions peuvent être différentes entre chaque occurrence du mot-clé `ECHEC` .

Si plusieurs causes d'échec sont satisfaites simultanément, l'ordre de déclenchement de ces échecs est prédéterminé :

`'ERREUR'` est prioritaire sur `'DELTA_GRANDEUR'`

3.2.1 Définition des causes d'échec

```
◊ EVENEMENT    =  /'ERREUR',              [DEFAUT]
                  /'DIVE_RESI'
                  /'DELTA_GRANDEUR'
                  /'COLLISION'
                  /'INTERPENETRATION'
                  /'INSTABILITE'
```

L'opérande `EVENEMENT` permet de définir les causes d'échec.

3.2.1.1 Événement `'ERREUR'`

Par défaut, la cause d'échec est `ERREUR` . Cette cause regroupe les raisons suivantes :

- Dépassement du nombre maximal autorisé d'itérations de Newton.
- Échec lors de l'intégration locale de la loi de comportement. Les paramètres locaux de l'algorithme utilisé pour l'intégration de la loi de comportement se trouvent dans [U4.51.11].
- Échec de l'algorithme de Deborst pour les contraintes planes ou les modèles 1D. On trouvera plus de détails dans [U4.51.11].
- Échec lors de la résolution de l'équation de pilotage (voir [U4.51.03]).
- Échec lors de la résolution du problème de contact discret (voir [R5.03.50]). Il y a échec lorsque le nombre maximum d'itérations de contact est dépassé ou lorsque la matrice de contact est singulière.
- La matrice du système est singulière. La détection de singularité s'appuie sur le paramètre `NPREC` du mot-clef facteur `SOLVEUR` (voir [U4.50.01]).
- Le solveur linéaire itératif a atteint le nombre maximum d'itérations de Krylov fixé par `NMAX_ITER` dans le mot-clef facteur `SOLVEUR` (voir [U4.50.01]).
- À convergence, des critères physiques ne sont pas satisfaits lors de l'intégration de la loi de comportement (notion d'« event-driven »)

Même si on ne trouve aucune occurrence de `ECHEC / EVENEMENT = 'ERREUR'` , le code en rajoute automatiquement une avec les paramètres actuels par défaut (c'est-à-dire l'action `'DECOUPE'` en mode `'MANUEL'` et avec `SUBD_NIVEAU=4`).

3.2.1.2 Événement `'DIVE_RESI'`

Le choix de l'événement `'DIVE_RESI'` permet de spécifier une autre cause possible d'échec (« event-driven »). Il y aura échec si le résidu ne diminue pas suffisamment au cours des itérations de Newton. Plus précisément, il y aura échec si : $min[R(i), R(i-1)] > R(i-2)$ pour $i \geq 3$, où $R(j)$ est le résidu (`RESI_GLOB_MAXI`) à l'itération de Newton j. S'il y a échec, alors le pas de temps est sous-découpé. L'intérêt ce cette fonctionnalité est qu'il n'est pas nécessaire d'attendre que l'on arrive à la fin des itérations de Newton autorisées pour re-découper le pas de temps en cas de non-convergence « notable ». On se rend compte de la non-convergence plus vite, un gain de temps est donc espéré.

Remarque :

L'activation de cette fonctionnalité avec le contact discret n'est pas conseillée car dans ce type de calcul le résidu ne décroît en général pas de façon monotone.

3.2.1.3 Événement `'DELTA_GRANDEUR'`

Le choix de l'événement `'DELTA_GRANDEUR'` permet de spécifier une autre cause possible d'échec (« event-driven »). Il y aura échec si l'incrément d'une composante d'un champ dépasse un seuil fixé, **à convergence**. Il faut alors définir le nom du champ par l'opérande `NOM_CHAM` (champ de déplacement, de contraintes ou de variables internes), le nom de la composante par l'opérande `NOM_CMP` et la valeur du seuil de référence par l'opérande `VALE_REF`.

```
# si EVENEMENT = 'DELTA_GRANDEUR'
    ♦ VALE_REF  =  valref ,              [R]
    ♦ NOM_CHAM  =  /'DEPL' ,
                   /'SIEF_ELGA',
                   /'VARI_ELGA',
    ♦ NOM_CMP   =  cmp,                  [TXT]
```

Plus précisément, soient v la composante `NOM_CMP` du champ `NOM_CHAM` et $valref$ la valeur de référence `VALE_REF`.

Si le champ `NOM_CHAM` est un champ aux nœuds et N_{no} est l'ensemble de tous les nœuds du maillage, alors il y aura échec si à convergence :

$$max(\Delta|v^i|, \forall i \in N_{no}) > valref$$

Si le champ `NOM_CHAM` est un champ aux points de Gauss, $N_{sp,pg,el}$ est l'ensemble de tous les sous-points des tous les points de Gauss de tous les éléments du modèle, alors il y aura échec si à convergence :

$$max(\Delta|v^i|, \forall i \in N_{sp,pg,el}) > valref$$

L'incrément Δ est à prendre au sens incrément entre deux instants.

Exemple n°1 avec une loi VMIS_ISOT_LINE :

```
ECHEC= _F(EVENEMENT     =  'DELTA_GRANDEUR',
          VALE_REF      =  0.1e-2,
          NOM_CHAM      =  'VARI_ELGA',
          NOM_CMP       =  'V1'),
```

On aura échec si l'incrément de la plasticité cumulée entre deux pas de temps dépasse 0,1%.

Exemple n°2 :

```
ECHEC=(
          _F(EVENEMENT     =  'DELTA_GRANDEUR',
             VALE_REF      =  5.e-2,
             NOM_CHAM      =  'DEPL',
             NOM_CMP       =  'DX',),
          _F(EVENEMENT     =  'DELTA_GRANDEUR',
             VALE_REF      =  5.e-2,
             NOM_CHAM      =  'DEPL',
             NOM_CMP       =  'DY'),
       )
```

On aura échec si la norme (norme sup) du déplacement (en 2D) varie de plus de 5e-2 entre deux pas de temps.

Remarque : dans les deux exemples précédents, bien que EVENEMENT='ERREUR' ne soit pas présent, il sera quand même activé en dur.

3.2.1.4 Événement 'COLLISION'

Cet événement permet de découper le pas de temps si le contact est détecté (impact). C'est une fonctionnalité particulièrement utile en dynamique. En effet, dans ce cas, une discrétisation temporelle inadaptée va faire apparaître des oscillations intempestives du déplacement et de la vitesse au moment du choc. Pour minimiser ces artefacts numériques, on recommande en général de découper finement le pas de temps au moment de la collision, ce qui n'est possible que pour les cas particuliers où cet instant est détectable *a priori* par l'utilisateur. Dans le cas général, on peut utiliser cette fonctionnalité pour que la découpe se fasse automatiquement.

3.2.1.5 Événement 'INTERPENETRATION'

♦ PENE_MAXI = penmax , [R]

Cet événement détecte l'interpénétration d'un nœud dans une surface lorsque le contact unilatéral de type régularisé (pénalisation) est activé. L'utilisateur donne une valeur de la pénétration maximale qu'il veut obtenir (paramètre PENE_MAXI) et le code adapte alors le coefficient de pénalisation (ACTION='ADAPT_COEF_PENA') ou s'arrête (ACTION='ARRET') pour respecter cette contrainte. Cette fonctionnalité n'est disponible que pour le contact en FORMULATION='DISCRETE' (voir DEFI_CONTACT [U4.44.11]).

3.2.1.6 Événement 'INSTABILITE'

Cet événement permet de gérer le déroulement d'un calcul non-linéaire lorsqu'une instabilité est détectée lors de l'appel à CRIT_STAB (*cf.* [U4.51.03] et [U4.53.01]). Si ce mot-clé CRIT_STAB n'est pas utilisé alors cet événement ne peut se produire.
Si une instabilité est détectée, une action sera alors déclenchée, suivant deux possibilités :

•ACTION='CONTINUE' : c'est la valeur par défaut et on va continuer le calcul, même en cas d'instabilité,
•ACTION='ARRET' : si une instabilité est captée, alors on va arrêter la résolution non-linéaire proprement (la base sera sauvegardée et réutilisable en poursuite).

Plus de détails sur le critère d'instabilité et son utilisation sont donnés dans les documentations [R7.05.01] et [U2.08.04].

3.2.2 Définition des actions à effectuer en cas d'échec

```
◊  ACTION =  /'DECOUPE',                         [DEFAUT]
              /'ARRET',
              /'ITER_SUPPL',
              /'AUTRE_PILOTAGE',
              /'ADAPT_COEF_PENA'
              /'CONTINUE'
```

Selon l'événement, seules certaines actions sont possibles, voir tableau ci-dessous :

ACTION ➞ / EVENEMENT ⬇	ARRET	DECOUPE	ITER_SUPPL	AUTRE_PILOTAGE	ADAPT_COEF_PENA	CONTINUE
ERREUR	OUI	OUI	OUI	OUI	NON	NON
DIVE_RESI	NON	OUI	NON	NON	NON	NON
DELTA_GRANDEUR	OUI	OUI	NON	NON	NON	NON
COLLISION	OUI	OUI	NON	NON	NON	NON
INTERPENETRATION	OUI	NON	NON	NON	OUI	NON
INSTABILITE	OUI	NON	NON	NON	NON	OUI

Par exemple, dans le cas le plus général (EVENEMENT='ERREUR'), il y a cinq possibilités :

- Découper le pas de temps (ACTION='DECOUPE') ;
- Autoriser des itérations de Newton supplémentaires (ACTION='ITER_SUPPL') ;
- Changer la solution issue du pilotage (ACTION='AUTRE_PILOTAGE') ;
- S'arrêter (ACTION='ARRET').

3.2.2.1 Action 'ARRET'

Si l'on choisit ACTION = 'ARRET', le calcul s'arrête proprement : la base est sauvegardée et sera ensuite réutilisable en poursuite.

3.2.2.2 Action 'DECOUPE'

```
◊ SUBD_METHODE   =  /'MANUEL'
                    /'AUTO'
# Si ACTION = 'DECOUPE' ET SI SUBD_METHODE = 'AUTO'
   ♦  SUBD_ INST      =   subd_inst,         [R]
   ♦  SUBD_ DUREE     =   SUBD_DUREE,        [R]

# Si SUBD_METHODE = 'MANUEL'
   ◊  SUBD_PAS        =  4.,              [DEFAUT]
                         pas,             [R]
   ◊  SUBD_NIVEAU     =  3 ,              [DEFAUT]
                         niv,             [I]
   ◊  SUBD_PAS_MINI   =  0.,              [DEFAUT]
                         pasmin,          [R]
# Si SUBD_METHODE = 'AUTO'
   ◊  SUBD_PAS_MINI   =  0.,              [DEFAUT]
                         pasmin,          [R]
```

Cette action consiste à refaire le pas de temps en le découpant. Il existe deux méthodes de découpe du pas de temps : le mode `AUTO` et le mode `MANUEL`.

Découpe en mode `MANUEL`

En mode `MANUEL`, le pas de temps est découpé de manière uniforme. Il faut donc préciser le nombre de découpes à l'aide du mot-clef `SUBD_PAS`.

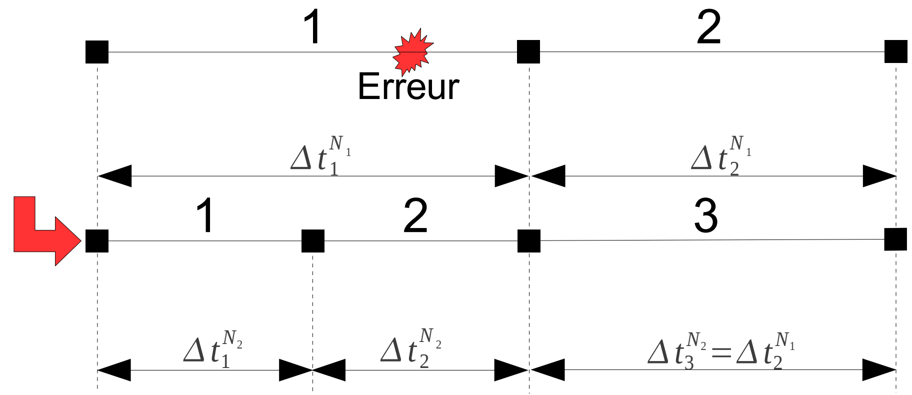


Figure 3-2: Découpe en mode `MANUEL` avec `SUBD_PAS=2`

On voit un exemple sur la figure 3-2. L'événement `ERREUR` se déclenche durant le pas de temps 1, initialement de longueur $\Delta t_1^{N_1}$. Ce pas est alors découpé en deux pas (`SUBD_PAS = 2`) de longueurs égales telles que $\Delta t_1^{N_2}=\Delta t_2^{N_2}=\frac{\Delta t_1^{N_1}}{2}$. Le pas suivant (numéro 2 dans la configuration initiale) devient alors le pas 3, sa longueur reste inchangée $\Delta t_3^{N_2}=\Delta t_2^{N_1}$.

Pas	1	2	3
Initial – 1	$\Delta t_1^{N_1}$	$\Delta t_2^{N_1}$	
Découpe – 2	$\Delta t_1^{N_2}$	$\Delta t_2^{N_2}$	$\Delta t_3^{N_2}$

La notion de niveau de découpe

La découpe du pas de temps est récursive : si l 'événement `ERREUR` se déclenche durant un pas déjà découpé, on redécoupe. On voit un exemple sur la figure 3-3.

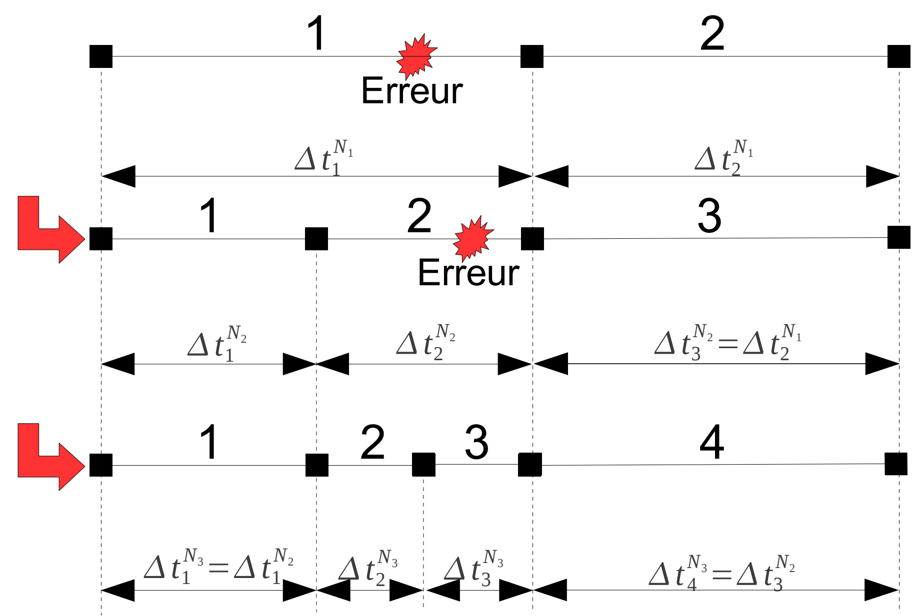


Figure 3-4: Découpe récursive en mode MANUEL avec SUBD_PAS=2

Après la première découpe, l'échec se produit dans le (nouveau) deuxième pas de temps. On redécoupe celui-ci avec $\Delta t_2^{N_3}=\Delta t_3^{N_3}=\dfrac{\Delta t_2^{N_2}}{2}$. Le pas de temps suivant reste inchangé $\Delta t_4^{N_3}=\Delta t_3^{N_2}$, le premier pas de temps est convergé et donc reste également inchangé $\Delta t_1^{N_3}=\Delta t_1^{N_2}$. Ce qui donne :

Pas	1	2	3	4
Initial – 1	$\Delta t_1^{N_1}$	$\Delta t_2^{N_1}$		
Découpe – 2	$\Delta t_1^{N_2}$	$\Delta t_2^{N_2}$	$\Delta t_3^{N_2}$	
Découpe – 3	$\Delta t_1^{N_3}$	$\Delta t_2^{N_3}$	$\Delta t_3^{N_3}$	$\Delta t_4^{N_3}$

Il ne peut y avoir un écart de plus d'un niveau entre deux pas de temps.

Découpe en mode AUTO

Le mode de découpe automatique utilise deux possibilités :

- Si l'événement déclencheur est une collision (`EVENEMENT='COLLISION'`), alors le pas de temps est découpé en ayant pour objectif d'avoir un pas de temps de longueur `subd_inst` , maintenu pendant la durée `subd_duree` . Si l'instant de découpe est trop près de la collision (contact trop “rasant”), la découpe ne sera pas efficace, il faut alors changer la discrétisation initiale ou le paramètre `subd_inst` (si le contact devient rasant après découpe) ;
- Sinon, on tente une extrapolation linéaire de la valeur des résidus.

Si l'on note $iter$ l'itération de Newton et $resi^{ext}$ la valeur du résidu extrapolé (`RESI_GLOB_RELA` et/ou `RESI_GLOB_MAXI`), alors on cherche a_1 , a_2 et b tels que :

$$resi^{ext}=\frac{(a_1.iter+a_2)}{b} \quad (1)$$

Voici l'algorithme utilisé :

```
Pour  iter=deb,fin
      xx=log(resi_{iter+1})
```

```
    Si  iter>fin−3
         poids=2
    Sinon
         poids=1
    On calcule
         xn=xn+poids
         sx=sx+poids.xx
         sy=sy+poids.iter
         sxx=sxx+poids.(xx^2)
         syx=syx+poids.xx.iter
Fin Pour
```

$a_1 = sxx.sy - sx.syx$

$a_2 = -(sx.sy) + syx.xn$

$b = -sx^2 + sxx.xn$

Pour faire cette extrapolation, on demande au minimum trois itérations de Newton et l'on considère que les deux dernières itérations ont deux fois plus de poids que les premières, si ce n'est pas possible (ou si l'extrapolation échoue pour une autre raison), on passe automatiquement en mode de découpe `MANUEL (SUBD_PAS=4)` .

Contrôle de la découpe

Si la convergence est difficile et que l'événement se déclenche plusieurs fois de suite, entraînant le découpage récursif du pas de temps, on peut arrêter le processus de deux manières en mode `MANUEL` :

- Avec le mot-clef `SUBD_NIVEAU`, le découpage s'arrêtera dès que le découpage sera fait plus de `niv` fois.
- Avec le mot-clef `SUBD_PAS_MINI` , le découpage s'arrêtera dès que le pas de temps sera inférieur à `pasmin` .

En mode `AUTO` , seul le mot-clef `SUBD_PAS_MINI` est disponible.

Remarque concernant le calcul de flambage :

Lors de calcul de flambage élastoplastique, il peut arriver que la matrice tangente du système soit singulière au cours des itérations de Newton. En re-découpant le pas de temps, on peut passer ces points durs.

Particularités de l'opérande `SUBD_NIVEAU` :

- L'opérande `SUBD_NIVEAU` est une notion qui devrait être globale à toutes les occurrences du mot-clé `ECHEC`. Ainsi, cet opérande ne devrait être défini qu'une seule fois. Dans la pratique, ce n'est pas le cas et rien n'empêche de définir une valeur de `SUBD_NIVEAU` différente à chaque occurrence du mot-clé `ECHEC`. Pour se prémunir de ce cas là, l'algorithme retiendra au final comme valeur globale et unique de `SUBD_NIVEAU` le **maximum** des `SUBD_NIVEAU` définis par l'utilisateur.

3.2.2.3 Action `'AUTRE_PILOTAGE'`

L'action `'AUTRE_PILOTAGE'` permet de définir une action dans un calcul qui fait appel au pilotage du chargement. Lors du pilotage, on peut avoir à choisir entre deux solutions (voir documentation [R5.03.80]) que l'on discrimine souvent à l'aide d'un critère `RESIDU`.

Si l'utilisateur sélectionne l'action `' AUTRE_PILOTAGE '` et que la convergence échoue avec le choix de solution qui minimise le résidu, on recommence le pas de temps courant, après l'avoir réinitialisé correctement, mais sans subdiviser. À la première itération où le choix se présente, on choisira la solution de résidu maximal lors de cette seconde tentative. On subdivise uniquement si les deux tentatives ont échoué.

Cette action sera donc entreprise uniquement si le nombre maximal d'itérations de Newton est atteint et s'il s'agit de la première tentative.

3.2.2.4 Action `'ITER_SUPPL'`

```
◊   PCENT_ITER_PLUS  =   50 ,                          [DEFAUT]
                         PCENT ,                       [I]
◊   SUBD_METHODE     =  /'MANUEL',                     [DEFAUT]
                        /'AUTO',
```

L'action `'ITER_SUPPL'` permet de faire des itérations de Newton au-delà du maximum donné par `ITER_GLOB_MAXI` (ou `ITER_GLOB_ELAS`). Elle évite que le calcul échoue pour quelques itérations de Newton. Pour cela, grâce à l'extrapolation linéaire des résidus (voir § 3.2.2.2), il est possible d'estimer le nombre d'itérations de Newton nécessaire à la convergence.

Le nombre d'itérations supplémentaires est limité par le mot-clé `PCENT_ITER_PLUS`. Si le nombre estimé d'itérations est supérieur à `pcent*ITER_GLOB_*`, alors on considère que l'on a échoué. De même si l'extrapolation ne réussit pas. Dans ces deux cas, il est possible d'activer la découpe du pas de temps (mot-clef `SUBD_METHODE`).

3.2.2.5 Action `'ADAPT_COEF_PENA'`

```
◊   COEF_MAXI    =   1E12                          [DEFAUT]
                     coemax,                       [R]
```

Si l'événement `INTERPENETRATION` se déclenche, cette action consiste à augmenter automatiquement la valeur du coefficient de pénalisation (voir opérateur `DEFI_CONTACT` [U4.44.11]) de manière à diminuer cette interpénétration. Le paramètre `COEF_MAXI` permet de spécifier la valeur maximale prise par le coefficient de pénalisation lorsque le mécanisme d'adaptation automatique est utilisé, ceci afin d'éviter des problèmes de conditionnement de la matrice (voir les précautions d'usage des méthodes pénalisées dans [U2.04.04]).

Actuellement le mécanisme ne fait qu'augmenter ce coefficient et ne le fait jamais diminuer, même si l'interpénétration n'apparaît plus après adaptation, il a donc tendance à transformer un problème de contact avec interpénétration en problème sans contact. Il convient donc de choisir un coefficient de pénalisation initial plus faible que nécessaire (voir opérateur `DEFI_CONTACT` [U4.44.11]).

3.2.2.6 Action `'CONTINUE'`

Si l'événement `INSTABILITE` se déclenche, cette action consiste à tenter de poursuivre le calcul non-linéaire. C'est la valeur par défaut pour cet événement, qui est le seul à pouvoir se combiner avec cette action.

3.3 Mot-clé `ADAPTATION`

L'adaptation du pas de temps consiste à calculer un nouveau pas de temps Δt^{i+1}, à partir des informations du pas de temps actuel Δt^{i} (et des pas de temps précédents Δt^{i-1}, Δt^{i-0}, ...).

Le but est donc de calculer $\Delta t^{i+1}=c.\Delta t^{i}$, où c est un coefficient réel.

Le procédé est le suivant :
Pour l'occurrence k du mot-clé `ADAPTATION`, on définit un événement et un mode de calcul de Δt^{i+1}. Si l'événement est vérifié, alors on calcule c^{k} avec le mode de calcul choisi.
On obtient un tableau récapitulatif de ce type :

Occurrence n°1	$c^{1}=4{,}2$

Occurrence n°2	événement non vérifié
Occurrence n°3	$c^3=1,7$
Occurrence n°4	$c^4=3,9$

Ensuite, on choisit $c=min(c^k)$ parmi les occurrences dont l'événement est vérifié. Dans cet exemple, le nouveau pas de temps vaudra $\Delta t^{i+1}=1,7.\Delta t^i$.
Pour utiliser MODE_CALCUL_TPLUS='IMPLEX', il faut que ce mot clé ne soit utilisé qu'une fois.

3.3.1 Opérande EVENEMENT

```
◊  EVENEMENT    =  /'SEUIL',                            [DEFAUT]
                   /'TOUT_INST'
                   /'AUCUN'
```

Cet opérande permet de spécifier le critère de déclenchement pour l'adaptation du pas de temps.

Si EVENEMENT='SEUIL', alors l'événement n'est vérifié que si le seuil est franchi.

Si EVENEMENT='TOUT_INST', alors l'événement est vérifié à tous les instants ; ce cas est obligatoire avec la méthode IMPLEX.

Si EVENEMENT='AUCUN', alors l'événement n'est jamais vérifié.

3.3.2 Opérandes NB_INCR_SEUIL, NOM_PARA, CRIT_COMP, VALE_I

```
◊  NB_INCR_SEUIL      =  /2,                       [DEFAUT]
                         /nbincseuil,              [I]
◊  NOM_PARA           =  'NB_ITER_NEWTON'          [DEFAUT]
◊  CRIT_COMP          =  /'LE',                    [DEFAUT]
                      =  /'LT',
                      =  /'GE',
                      =  /'GT',
◊  VALE_I             =   nbvalseuil,              [I]
```

Ces opérandes ne sont à renseigner qu'en cas d'événement de type seuil (EVENEMENT='SEUIL '). L'événement est vérifié si on a nbincseuil fois de suite un « feu vert », un feu vert étant caractérisé par la condition NOM_PARA CRIT_COMP VALE_I.

Par exemple si NB_INCR_SEUIL=2 , NOM_PARA='NB_ITER_NEWTON' , CRIT_COMP=LE' et VALE_I=5 , alors l'événement est vérifié si on a fait 2 fois de suite moins de 5 itérations de Newton.

Remarque :
Il n'y a pas de valeur par défaut pour VALE_I dans le catalogue de l'opérateur DEFI_LIST_INST . Cependant, si sa valeur n'est pas renseignée, on prendra pour VALE_I la moitié du nombre maximal d'itérations de Newton (déclaré dans STAT_NON_LINE/CONVERGENCE/ITER_GLOB_MAXI).

3.3.3 Opérande MODE_CALCUL_TPLUS

```
 ◊ MODE_CALCUL_TPLUS   =  /   'FIXE',                       [DEFAUT]
                          /   'DELTA_GRANDEUR'
                          /   'ITER_NEWTON'
                          /   'IMPLEX'
```

Cet opérande sert à spécifier le mode de calcul du nouveau pas de temps (en fait, du coefficient c).

Si `MODE_CALCUL_TPLUS='FIXE'`, alors le coefficient c est fixe. Sa valeur est déterminée grâce à l'opérande `PCENT_AUGM`. Le nouveau pas de temps vaudra alors $\Delta t^{i+1} = (1 + \frac{\text{pcent_augm}}{100.}) \Delta t^{i}$.

Si `MODE_CALCUL_TPLUS='DELTA_GRANDEUR'`, alors le coefficient c est lié à la variation au cours du pas de temps d'une grandeur choisie. On utilise la formule suivante :
$c = \min_{i \in N} \left(\frac{V_{ref}}{|\Delta V_i|} \right)$ avec N l'ensemble des nœuds ou des points de Gauss pour lesquels ΔV_i n'est pas nul. La valeur de référence V_{ref} est choisie par l'utilisateur.
Si le champ ΔV est identiquement nul, alors on considère que le critère (l'événement) n'est pas vérifié.

Si `MODE_CALCUL_TPLUS='ITER_NEWTON'`, alors le coefficient c est lié à la variation au cours du pas de temps du nombre d'itérations de Newton. On utilise la formule suivante :
$c = \sqrt{\frac{N_{ref}}{N_{it} + 1}}$ où N_{it} est le nombre d'itérations de Newton effectuées et N_{ref} une valeur de référence choisie par l'utilisateur. On rappelle que la prédiction est considérée comme l'itération de Newton n°0, ce qui signifie que si on converge dès la prédiction alors $c = \sqrt{N_{ref}}$.

Si `MODE_CALCUL_TPLUS='IMPLEX'`, alors le coefficient c est lié aux incréments de variables internes (déformation plastique cumulée ou endommagement) du pas précédent (voir [R5.03.81]). Dans ce cas, le mot clé `ADAPTATION` ne doit être utilisé qu'une seule fois. Le coefficient c sera forcément compris entre 0,5 et 1,2. De plus, par définition de la méthode `IMPLEX` il n'y aura aucune cause d'échec.

3.3.4 Opérande `PCENT_AUGM`

```
◊  PCENT_AUGM   =  /100.,                              [DEFAUT]
                   /pcent ,                            [R]
```

Cet opérande sert à spécifier le pourcentage d'augmentation du pas de temps si `MODE_CALCUL_TPLUS = 'FIXE'`. Par défaut, on augmente de 100%, c'est-à-dire que l'on double la valeur du pas de temps. La valeur peut être négative et doit être strictement supérieure à -100.

3.3.5 Opérandes `VALE_REF, NOM_CHAM, NOM_CMP`

```
♦  VALE_REF     =  valref,                             [R]
♦  NOM_CHAM     =  /'DEPL',
                   /'SIEF_ELGA',
                   /'VARI_ELGA',
♦  NOM_CMP      =  cmp,                                [K8]
```

Ces opérandes servent à spécifier le calcul du coefficient c d'augmentation du pas de temps si `MODE_CALCUL_TPLUS = 'DELTA_GRANDEUR'` (cf. §3.3.3) avec `VALE_REF` la valeur de référence V_{ref} et `NOM_CHAM` le nom du champ de grandeur considéré.

3.3.6 Opérande `NB_ITER_NEWTON_REF`

```
♦  NB_ITER_NEWTON_REF  =  nitref,                      [I]
```

Cet opérande sert à renseigner le nombre d'itérations de Newton de référence, noté N_{ref} (cf. §3.3.3) dans le cas où `MODE_CALCUL_TPLUS = 'ITER_NEWTON'`.

3.4 Opérande `INFO`

```
◊  INFO
```

Pour le moment, cet opérande ne sert pas.